

BOSS: A FORTRAN CODE FOR A RELATIONAL DATABASE MANAGER
(U) NAVAL SURFACE WEAPONS CENTER SILVER SPRING MD
E WINSTON 01 MAY 85 NSWC/TR-85-56

UNCLASSIFIED

F/G 9/2

NL

Page 1 of 1

1 2

1 2

END

1 2



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A157 067

2

NSWC TR 85-56

**BOSS:
A FORTRAN CODE FOR A RELATIONAL
DATABASE MANAGER**

BY ELLIOT WINSTON

RESEARCH AND TECHNOLOGY DEPARTMENT

1 MAY 1985

Approved for public release; distribution is unlimited.

DTIC
ELECTE
JUL 26 1985
S B D



NAVAL SURFACE WEAPONS CENTER

Dahlgren, Virginia 22448 • Silver Spring, Maryland 20910

85 7 16 027

DTIC FILE COPY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NSWC TR 85-56	2. GOVT ACCESSION NO. AD-A157067	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) BOSS: A FORTRAN CODE FOR A RELATIONAL DATABASE MANAGER		5. TYPE OF REPORT & PERIOD COVERED Final; Fiscal Year 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Elliot Winston		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Surface Weapons Center (Code R44) 10901 New Hampshire Avenue Silver Spring, MD 20903-5000		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 64601N; S0267; 0; 5U15DD
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 1 May 1985
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 97
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Relational Database Manager B+ Tree		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Instruction and documentation for an interactive relational database manager is presented, based on a B+ tree data structure for rapidly retrieving record keys.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This report contains documentation for a FORTRAN implementation of a relational database manager. Because the code is written in a high-level language, it is basically transportable to any computer with FORTRAN capability (minor modification may be required for compatability with a host computer's operating system and compiler). The work was required by U31 to support computer studies requiring the extensive use of minefield planning codes.

This work has been supported by the Mine Improvement Program at NSWC under Project S0267.

Approved by:

Ira M. Blatstein

IRA M. BLATSTEIN, Head
Radiation Division

DTIC
ELECTE
S **D**
JUL 26 1985
B

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



CONTENTS

	<u>Page</u>
INTRODUCTION.....	1
DATA CATEGORIES.....	1
CATEGORY CREATION.....	2
CATEGORY REQUESTS.....	3
EXAMPLE.....	3
RECONFIGURATION.....	5
APPLICATION PROGRAMS.....	6
REFERENCES.....	7
APPENDIX A - SUBROUTINE DOCUMENTATION.....	A-1
APPENDIX B - FORTRAN CODE LISTING.....	B-1
DISTRIBUTION.....	(1)

TABLES

<u>Table</u>	<u>Page</u>
1 COURSES.....	4
2 FACULTY.....	4
3 ASSIGN.....	5
4 ASSIGN.....	5

INTRODUCTION

This report contains instruction and documentation for an interactive relational database manager code called BOSS. BTREE, which is an implementation of a B+ tree and is documented in Winston,¹ provides the fundamental data structure utilized by BOSS for rapidly retrieving data record keys. This work completes the effort begun with BTREE to develop a user-friendly code to manage and maintain medium-sized databases, thereby providing U31 with the capability to efficiently and easily perform large-scale computer studies which analyze various questions related to minefield planning.

The following sections constitute a manual for using BOSS, along with an illustrative example; Appendix A contains documentation of the subprograms and Appendix B contains a complete listing of the code itself. The code is written in a DEC version of FORTRAN 77 for a VAX/VMS system, and is therefore essentially transportable to any computer with FORTRAN capability. (Minor modification may be required for compatibility with a host computer's operating system and compiler.) The format for file names is assumed to be (name).(ext), where (name) consists of at most 9 characters, and (ext) is an extender, or modifier, of at most 3 characters.

DATA CATEGORIES

BOSS can manage several logically independent collections of data, henceforth called categories. A data record in a category consists of a number of fields, each of which is described by a set of parameters: field name, data type, field length, and resource category.

- 1) The field name is usually chosen to be a generic descriptor of the data stored in the field.
- 2) A discussion of each data type follows:
 - (a) "Character data" is data which generally consists of names and descriptive words, but can also be a string of numbers, usually interspersed by separators for parsing and conversion into actual numerical value by an application program. (This is a convenient way to store a row or column in a numerical table.) BOSS, itself, never ascribes any numerical significance to such data.
 - (b) "Numerical data" differs from character data in that the user may request BOSS to compare it with respect to its numerical, and not lexicographical, value.
 - (c) The function ENDATE converts a date between January 1, 1900 and December 31, 2075 into the number of days since December 31, 1899. The function SYM then uses ASCII symbols to convert this integer into a 2-byte symbol. (The inverse process is accomplished by calling VAL and DEDATE.) Hence, only 2 bytes of memory are required

to store such a date, a much smaller memory requirement than interpreting the date as character data. Moreover, a comparison between different dates is easily accomplished by comparing their associated integer values. Character data provides an adequate way of storing dates outside the allowable range.

- (d) The category TABLE is a special category created by BOSS to save mass storage memory and also aid in reducing keystroke errors. When a field has a limited number of possible values, e.g., color, job title, etc., it is more efficient to enter each of the possible values once as data in TABLE, and instead store the associated TABLE record number, or pointer, in the corresponding field of the actual data record. As previously discussed above in part (c), a pointer requires only 2 bytes of memory. Thus, records in TABLE consist of two fields: the field name, and the field value, both designated to have a field length of 10 characters. Also, when data records are added to the current category, all of the possible values of any field with "table data" are displayed in a numbered list from which the user makes a selection, thereby eliminating the burden of entering the complete data value.
 - (e) "Duplicate data" is somewhat similar to table data in that pointers are stored rather than actual data values. If the data records of different categories contain a common field, that is, a field with the same name and same set of data values, it may be possible to avoid a complete duplication of the field in each of the categories. If the field in question is the key for some category, then pointers can be stored as the field data in the other categories containing the field. (See the next section for an explanation of keys.) It is important to stress that duplicate fields in different categories must have exactly the same field name!
- 3) The field length is the maximum number of characters required by any of the possible field data values.
 - 4) The resource category is the name of the category containing the actual field data values rather than any associated pointers.

CATEGORY CREATION

In order to create a category, a category name, a category password (optional), the number of fields in a typical category data record, and the number of the key field all must be supplied by the user. The data in the key field is called the key and must uniquely identify the data record. These four category parameters are stored in the file CAT.DAR, a record of which uses the category name as its key; CAT.KEY is the associated B+ tree. In addition, the user must supply the field parameters, discussed in the previous section, which describe a typical category data record. The parameters for the n th field are contained in the n th record of (name).LAR.

When appropriate, parameters have default values assigned to them by BOSS. For example, the field length is automatically set equal to 2 when the data type of a field is neither character nor numerical. All information which must be supplied by the user is entered in response to a series of prompts by BOSS.

The entire collection of data record keys is stored in the B+ tree (name).KEY, and (name).DAR contains the associated category data records.

The specifications of the implementation of BOSS given in Appendix B are:

maximum number of records per category	65,535
maximum number of fields per record	20
maximum number of characters per field	100
maximum number of characters per record	256
maximum number of categories associated with a current category via duplicate data	7

CATEGORY REQUESTS

Most category requests are self-explanatory, such as adding, getting, deleting, or modifying a category data record. In addition, the user can change the category password, review the record field parameters, inquire about the number of records currently in a category, or write all the records in a category to an output file. A special type of search, called a "range query", retrieves all the records in a category which satisfy a particular set of conditions. The user selects a subset of all the record fields, and for each such field, specifies a range of values within which data in that field must lie. Since BOSS examines every record in a category to execute this request, it is possible for this procedure to consume a relatively larger amount of time.

EXAMPLE

The example discussed in this section is purely hypothetical, but is useful in demonstrating how to specify the parameters needed to define categories. The more fundamental problem of identifying which collections of data are appropriate as categories is not addressed in this report,³ and therefore the reader is urged to consult Kent² or Neely and Steward³ for very readable introductions to the important concepts of logically independent data and normal⁴ forms. A more theoretical discussion can be found in Stout and Woodworth.⁴

Table 1 is a list of all the courses offered by a small mathematics department. This table of data constitutes the category "COURSES".

TABLE 1. COURSES

Title	Number	Credits
Calculus I	120	4
Calculus II	121	4
Linear Algebra	235	3
Probability	250	3
Statistics	251	3
Analysis	310	3

The second field serves as the key field, and is admissible as the key because the course number uniquely identifies all the data in the row (record) containing it. The first field, "Title", also qualifies as a key, but is not as convenient for defining the category "ASSIGN", below. "Title" is assumed to contain character data with a field length of 15 characters, "Number" has numerical data with a field length of 3, and "Credits" also has numerical data, but with a field length of 1. The data type of "Number" is chosen to be numerical to give the user the ability to make certain types of requests, such as asking for a list of all 200-level courses. This can be accomplished by a range query on field 2 with an inclusive upper bound of 299 and an inclusive lower bound of 200.

Table 2 is a faculty list. The key field, "Name", has character data

TABLE 2. FACULTY

Name	Rank
Jones	lecturer
Smith	instructor
Brown	instructor
Thomas	professor
Johnson	instructor

with a field length of 10. (Although 7 characters are sufficient for all current faculty names, the field is defined to be a bit larger to allow for possible future changes in faculty.) Since there are only three faculty ranks, the second field, "Rank", is assumed to contain table data. The three associated records in "Table" are (rank,lecturer), (rank,instructor), and (rank,professor).

Finally, the course assignments listed in Table 3 provide the data for the category "ASSIGN". Fields 1 and 3 are copies of key fields in other

TABLE 3. ASSIGN

Number	Section	Name
120	1	Jones
120	2	Johnson
121	1	Jones
121	2	Johnson
235	1	Smith
250	1	Brown
251	1	Brown
312	1	Thomas

categories. Consequently, they are assumed to have duplicate type data which is related to the resource categories "Courses" and "Faculty"; the data type of "Section" is numerical with a field length of 1. However, no single field can serve as the key field because, in general, no row is uniquely identified by the data in any one field. The data in field 1 together with the data in field 2 do identify rows, and thus, an additional field containing "compound" data is added to the category, as is shown in Table 4.

TABLE 4. ASSIGN

Key	Number	Section	Name
1201	120	1	Jones
1202	120	2	Johnson
1211	121	1	Jones
1212	121	2	Johnson
2351	235	1	Smith
2501	250	1	Brown
2511	251	1	Brown
3101	310	1	Thomas

The new field, "Key", is an artifice which provides "ASSIGN" with a key. This device is not uncommon in practice.

RECONFIGURATION

The following specifications can be altered easily to satisfy special requirements of the user:

- (a) To change the maximum number of fields per record to *f*, declare the arrays LONG(*f*), IO(*f*), TYPE(*f*), FLD(*f*), TITLE(*f*), WHERE(*f*), INA(*f*), INB(*f*), EXA(*f*), and EXB(*f*) in COMMON/XXXBOSS/, and LINK(*f*) and WIDTH(*f*) in SUBROUTINE OUTPUT;

- (b) To change the maximum number of characters per field to *c*, declare the array `FLD(f)` as `CHARACTER*c` (the maximum key length in `BTREE` should also be checked to make sure it is at least as big as *c*);
- (c) To change the maximum number of characters per record to *r*, declare the variable `RECDATA` as `CHARACTER*r`.

APPLICATION PROGRAMS

In order for an application program to retrieve data from one or more categories, the user need only check the source code of `BOSS` to find out how to access a category and its data. Usually, this requires little more than adding `SUBROUTINE OPENCAT` and `SUBROUTINE RECOUNT` to the application program, and writing a short subroutine to get the appropriate data. Of course, the application program must also be linked with `BTREE` when forming the executable image.

REFERENCES

1. Winston, E., BTREE: A FORTRAN Code for a B+ Tree, NSWC TR 85-54, Apr 1985.
2. Kent, W., "A Simple Guide to Five Normal Forms in Relational Database Theory," Communications ACM, Vol 26, No. 2, 1983, pp. 120-126.
3. Neely, J., and Steward, S., "Fundamentals of Relational Data Organization," BYTE, Nov 1981, pp. 51-60.
4. Stout, Q., and Woodworth, P., "Relational Databases," MAA Monthly, Vol. 90, 1983, pp. 101-118.

NSWC TR 85-56

APPENDIX A
SUBROUTINE DOCUMENTATION

SUBROUTINE MODWORD

PURPOSE: To modify the current category password.

INPUTS:

CATNAME	CHARACTER*9	name of the current category
NINE	BYTE	parameter set equal to 9
HOW()	CHARACTER*6	array of record formats

OUTPUTS:

none

EXTERNALS:

BTREE

SUBROUTINE MODREC

PURPOSE: To control the logic for modifying a data record.

INPUTS:

CATNAME	CHARACTER*9	name of the current category
KEYFLD	INTEGER*4	number of the key field
INA()	BYTE	array of pointers for start of each field in current category record
INB()	BYTE	array of pointers for end of each field in current category record
ONE,NINE	BYTE	parameter set equal to 1,9
HOW()	CHARACTER*6	array of record formats

OUTPUTS:

none

EXTERNALS:

FETCH,RECOUT,VERIFY,BTREE,RECIN,INSERT

SUBROUTINE DELREC

PURPOSE: To control the logic for deleting a data record.

INPUTS:

CATNAME	CHARACTER*9	name of the current category
NFIELD	INTEGER*4	number of current fields
KEYFLD	INTEGER*4	number of the key field
TITLE()	CHARACTER*10	array of field names
ONE,NINE	BYTE	parameter set equal to 1,9
HOW()	CHARACTER*6	array of record formats

OUTPUTS:

none

EXTERNALS:

BTREE,OPENCAT,VAL,RECOUT,SHOWREC,RECIN

SUBROUTINE FETCH

PURPOSE: To retrieve a data record in the current category.

INPUTS:

CATNAME	CHARACTER*9	name of the current category
KEYFLD	INTEGER*4	number of the key field
TITLE()	CHARACTER*10	array of field names
TYPE()	INTEGER*4	array of field data types
WHERE()	CHARACTER*9	array of resource categories
IO()	BYTE	array of unit number links between fields and resource categories
HOW()	CHARACTER*6	array of record formats
ONE	BYTE	parameter set equal to 1

OUTPUTS:

RECDATA	CHARACTER*256	data record in current category
---------	---------------	---------------------------------

EXTERNALS:

SYM, ENDATE, TABLIST, BTREE

SUBROUTINE GETREC

PURPOSE: To control the logic for getting a data record.

INPUTS:

CATNAME	CHARACTER*9	name of current category
ONE	BYTE	parameter set equal to 1
HOW()	CHARACTER*6	array of record formats

OUTPUTS:

none

EXTERNALS:

FETCH, RECOUT, SHOWREC

SUBROUTINE INSERT

PURPOSE: To insert a data record into the current category.

INPUTS:

CATNAME	CHARACTER*9	name of the current category
NFIELD	INTEGER*4	number of current fields
KEYFLD	INTEGER*4	number of the key field
FLD()	CHARACTER*100	array of field data in internal format
INA()	BYTE	array of pointers for start of each field in current category record
INB()	BYTE	array of pointers for end of each field in current category record
ONE	BYTE	parameter set equal to 1
HOW()	CHARACTER*6	array of record formats

OUTPUTS:

none

EXTERNALS:

BTREE

SUBROUTINE VERIFY

PURPOSE: To verify a data record.

INPUTS:

NFIELD	INTEGER*4	number of current fields
FLD()	CHARACTER*100	array of field data in external format
TITLE()	CHARACTER*10	array of field names

OUTPUTS:

FLD()	CHARACTER*100	updated array of field data in external format
--------	---------------	--

EXTERNALS:

SHOWREC,CHECK,TABLIST

SUBROUTINE ADDREC

PURPOSE: To control the logic for adding a record.

INPUTS:

NFIELD	INTEGER*4	number of current fields
KEYFLD	INTEGER*4	number of the key field
TITLE()	CHARACTER*10	array of field names
TYPE()	INTEGER*4	array of field data types
EXA()	BYTE	array of pointers for start of duplicate fields in resource category records
EXB()	BYTE	array of pointers for end of duplicate fields in resource category records

OUTPUTS:

none

EXTERNALS:

TABLIST, VERIFY, RECIN, INSERT

SUBROUTINE DELCAT

PURPOSE: To control the logic for deleting a category.

INPUTS:

CATNAME	CHARACTER*9	name of the current category
HOW(9)	CHARACTER*6	record format for CAT.DAR
HOW(10)	CHARACTER*6	record format for (CATNAME).LAR

OUTPUTS:

none

EXTERNALS:

BTREE

SUBROUTINE OPENCAT

PURPOSE: To initialize the parameters of the current category.

INPUTS:

HOW(10)	CHARACTER*6	record format for (CATNAME).LAR
CATNAME	CHARACTER*9	name of the current category
NFIELD	INTEGER*4	number of current fields

OUTPUTS:

IO()	BYTE	array of unit number links between fields and resource categories
TITLE()	CHARACTER*10	array of field names
TYPE()	INTEGER*4	array of field data types
LONG()	INTEGER*4	array of field lengths
WHERE()	CHARACTER*9	array of resource categories
INA()	BYTE	array of pointers for start of each field in current category record
INB()	BYTE	array of pointers for end of each field in current category record
EXA()	BYTE	array of pointers for start of duplicate fields in resource category records
EXB()	BYTE	array of pointers for end of duplicate fields in resource category records
NREF	BYTE	number of related categories with respect to current category
HOW()	CHARACTER*6	array of record formats

EXTERNALS:

BTREE

SUBROUTINE PICKCAT

PURPOSE: To either delete a category selected by the user,
or open a category and execute category requests.

INPUTS:

HOW(9)	CHARACTER*6	<i>record format for CAT.DAR</i>
--------	-------------	----------------------------------

OUTPUTS:

CATNAME	CHARACTER*9	name of the current category
KEYFLD	INTEGER*4	number of the key field
NFIELD	INTEGER*4	number of current fields

EXTERNALS:

BTREE,CHECK,TABMENU,DELCAT,OPENCAT,ADDREC,GETREC,DELREC,MODREC,
QUERY,CATLIST,MODWORD,REVIEW

SUBROUTINE VIEWSPEC

PURPOSE: To review the field parameters of the current category; editing permitted during category creation only.

INPUTS:

HOW(10)	CHARACTER*6	record format for (CATNAME).LAR
CATNAME	CHARACTER*9	name of the current category
KEYFLD	INTEGER*4	number of the key field
TITLE()	CHARACTER*10	array of field names
TYPE()	INTEGER*4	array of field data types
LONG()	INTEGER*4	array of field lengths
WHERE()	CHARACTER*9	array of resource categories
NEW	LOGICAL*1	.TRUE. upon category creation, .FALSE. otherwise

OUTPUTS:

TITLE()	CHARACTER*10	array of field names
TYPE()	INTEGER*4	array of field data types
LONG()	INTEGER*4	array of field lengths
WHERE()	CHARACTER*9	array of resource categories

EXTERNALS:

CHECK, SYM

SUBROUTINE NEWCAT

PURPOSE: To create a new category.

INPUTS:

HOW(9)	CHARACTER*6	record format for CAT.DAR
HOW(10)	CHARACTER*6	record format for (CATNAME).LAR

OUTPUTS:

CATNAME	CHARACTER*9	name of the current category
NFIELD	INTEGER*4	number of current fields
KEYFLD	INTEGER*4	number of the key field
MAXLEN	INTEGER*4	length of the key field
TITLE()	CHARACTER*10	array of field names
TYPE()	INTEGER*4	array of field data types
WHERE()	CHARACTER*9	array of resource categories
LONG()	INTEGER*4	array of field lengths
NEW	LOGICAL*1	.TRUE. upon category creation, .FALSE. otherwise

EXTERNALS:

CHECK, SYM

PROGRAM BOSS

PURPOSE: To open the database and control the logic needed to execute user requests.

INPUTS:

none

OUTPUTS:

HOW(9) CHARACTER*6 record format for CAT.DAR

HOW(10) CHARACTER*6 record format for (CATNAME).LAR

EXTERNALS:

BTREE,CHECK,PICKCAT,NEWCAT

SUBROUTINE REVIEW

PURPOSE: To select a field and review its parameters.

INPUTS:

 none

OUTPUTS:

 none

EXTERNALS:

 FLDLIST,VIEWSPEC

SUBROUTINE FLDLIST

PURPOSE: To list the field names of the current category and select one of them.

INPUTS:

CATNAME	CHARACTER*9	name of the current category
NFIELD	INTEGER*4	number of current fields
TITLE()	CHARACTER*10	array of field names

OUTPUTS:

N	INTEGER*4	number of selected field
---	-----------	--------------------------

EXTERNALS:

CHECK

SUBROUTINE RECIN

PURPOSE: To transform a record from external format into internal format.

INPUTS:

NFIELD	INTEGER*4	number of current fields
FLD()	CHARACTER*100	array of field data in external format
TYPE()	INTEGER*4	array of field data types
TITLE()	CHARACTER*10	array of field names
WHERE()	CHARACTER*9	array of resource categories
IO()	BYTE	array of unit number links between fields and resource categories
TEN	BYTE	parameter set equal to 10

OUTPUTS:

FLD()	CHARACTER*10	array of field data in internal format
--------	--------------	--

EXTERNALS:

SYM, ENDATE, BTREE

SUBROUTINE RECOU

PURPOSE: To transform a record from internal format into external format.

INPUTS:

NFIELD	INTEGER*4	number of current fields
INA()	BYTE	array of pointers for start of each field in current category record
INB()	BYTE	array of pointers for end of each field in current category record
RECDATA	CHARACTER*256	data record in current category
TEN	BYTE	parameter set equal to 10
IO()	BYTE	array of unit number links between fields and resource categories
HOW()	CHARACTER*6	array of record formats
TITLE()	CHARACTER*10	array of field names
TYPE()	INTEGER*4	array of field data types
EXA()	BYTE	array of pointers for start of duplicate fields in resource category records
EXB()	BYTE	array of pointers for end of duplicate fields in resource category records

OUTPUTS:

FLD()	CHARACTER*10	array of field data in external format
--------	--------------	--

EXTERNALS:

VAL,BTREE

SUBROUTINE SHOWREC

PURPOSE: To display a record on the screen.

INPUTS:

NFIELD	INTEGER*4	number of current fields
TITLE()	CHARACTER*10	array of field names
FLD()	CHARACTER*10	array of field data in external format

OUTPUTS:

none

EXTERNALS:

none

SUBROUTINE QUERY

PURPOSE: To control the logic for a range query.

INPUTS:

HOW()	CHARACTER*6	array of record formats
TYPE()	INTEGER*4	array of field data types
ONE,SIX	BYTE	parameter set equal to 1,6

OUTPUTS:

none

EXTERNALS:

FLDLIST,CHECK,SYM,ENDATE,BTREE,RECOUT,VAL,CONVERT,OUTPUT

SUBROUTINE CATLIST

PURPOSE: To write the number of every record of the
current category on a scratch file.

INPUTS:

none

OUTPUTS:

none

EXTERNALS:

BTREE,OUTPUT

SUBROUTINE OUTPUT

PURPOSE: To write a set of records on an output file.

INPUTS:

CATNAME	CHARACTER*9	name of the current category
NFIELD	INTEGER*4	number of current fields
TITLE()	CHARACTER*10	array of field names
TYPE()	INTEGER*4	array of field data types
EXA()	BYTE	array of pointers for start of duplicate fields in resource category records
EXB()	BYTE	array of pointers for end of duplicate fields in resource category records

OUTPUTS:

none

EXTERNALS:

FLDLIST, RECOU

SUBROUTINE TABMENU

PURPOSE: To control the logic for a "TABLE" request.

INPUTS:

HOW()	CHARACTER*6	array of record formats
FIVE,NINE	BYTE	parameter set equal to 5,9

OUTPUTS:

none

EXTERNALS:

BTREE,CHECK,TABLIST,TABDEL,TABADD

SUBROUTINE TABADD

PURPOSE: To add a record to "TABLE".

INPUTS:

FLDNAME	CHARACTER*10	name of table field
TEN	BYTE	parameter set equal to 10

OUTPUTS:

none

EXTERNALS:

BTREE

SUBROUTINE TABDEL

PURPOSE: To delete a record from "TABLE".

INPUTS:

FLDVAL	CHARACTER*10	value of table field
FLDNAME	CHARACTER*10	name of table field
ONE,NINE,TEN	BYTE	parameter set equal to 1,9,10
HOW()	CHARACTER*6	array of record formats
NREF	BYTE	number of related categories with respect to current category

OUTPUTS:

none

EXTERNALS:

BTREE,VAL,OPENCAT,RECOUT,SHOWREC

SUBROUTINE TABLIST

PURPOSE: To list all the current values of a field with table type data.

INPUTS:

FLDNAME	CHARACTER*10	name of table field
TEN	BYTE	parameter set equal to 10
IND	INTEGER*4	indicator which selects appropriate screen message

OUTPUTS:

FLDVAL	CHARACTER*10	value of table field
--------	--------------	----------------------

EXTERNALS:

BTREE,CHECK

SUBROUTINE CHECK

PURPOSE: To trap a particular class of typographical error.

INPUTS:

ANS	CHARACTER*3	numerical user input in string format
NMAX	INTEGER*4	largest admissible value for user input

OUTPUTS:

N	INTEGER*4	integer value of user input
---	-----------	-----------------------------

EXTERNALS:

none

FUNCTION ENDATE

PURPOSE: To convert the date passed by WHEN into the number of days since December 31,1899.

INPUTS:

WHEN CHARACTER*10 date in string format

OUTPUTS:

ENDATE INTEGER*4 number of days since December 31,1899 associated with WHEN

EXTERNALS:

none

```

HOW(1) = '(A'//HOW(1)(1:K)//')'
NREF = 1
DO 4040 I=1,NFIELD
  IF (TYPE(I).LE.2) THEN
    EXA(I) = INA(I)
    EXB(I) = INB(I)
  ELSE IF (TYPE(I).EQ.4) THEN
    IO(I) = 10
  ELSE IF (TYPE(I).EQ.5) THEN
    NREF = NREF + 1
    IO(I) = NREF
    LTR = 'O'
    A = WHERE(I)
    CALL BTREE(LTR,NREF,A,MAXLEN,IREC,IERR)
    LDU = NREF + 10
    CLOSE(UNIT=LDU)
    OPEN(UNIT=LDU,FILE=WHERE(I)//'.DAR',STATUS='OLD',
*      FORM='FORMATTED',ACCESS='DIRECT')
    LTR = 'G'
    CALL BTREE(LTR,NINE,A,MAXLEN,IREC,IERR)
    READ(19,HOW(9),REC=IREC) RECDATA
    MEND = VAL(RECDATA(20:21))
    CLOSE(UNIT=20)
    OPEN(UNIT=20,FILE=WHERE(I)//'.LAR',STATUS='OLD',
*      FORM='FORMATTED',ACCESS='DIRECT')
    DO 4020 M=1,MEND
      READ(20,HOW(10),REC=M) RECDATA
      IF (M.EQ.1) THEN
        MA = 1
        MB = VAL(RECDATA(11:12))
      ELSE
        MA = MB + 1
        MB = MB + VAL(RECDATA(11:12))
      END IF
      IF (TITLE(I).EQ.RECDATA(1:10)) THEN
        EXA(I) = MA
        EXB(I) = MB
      END IF
4020    CONTINUE
      K = 1
      DO WHILE (MB/10**K.GT.0)
        K = K + 1
      END DO
      ENCODE(K,402,HOW(NREF)) MB
      HOW(NREF) = '(A'//HOW(NREF)(1:K)//')'
    END IF
4040 CONTINUE
  RETURN
END

```

SUBROUTINE OPENCAT

C

IMPLICIT INTEGER*4 (A-Z)

C

COMMON /XXXBOSS/

```

1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

```

C

402 FORMAT(I<K>)

C

C-----

C

INITIALIZE CATEGORY PARAMETERS

C-----

C

LTR = '0'

A = CATNAME

CALL BTREE(LTR,ONE,A,MAXLEN,IERR,IERR)

CLOSE(UNIT=11)

OPEN(UNIT=11,FILE=CATNAME//'.DAR',STATUS='OLD',FORM='FORMATTED',

* ACCESS='DIRECT')

CLOSE(UNIT=20)

OPEN(UNIT=20,FILE=CATNAME//'.LAR',STATUS='OLD',FORM='FORMATTED',

* ACCESS='DIRECT')

LENREC = 0

DO 4005 I=1,NFIELD

IO(I) = 1

READ(20,HOW(10),REC=I) RECDATA

TITLE(I) = RECDATA(1:10)

LONG(I) = VAL(RECDATA(11:12))

LENREC = LENREC + LONG(I)

IF (I.EQ.1) THEN

INA(I) = 1

INB(I) = LONG(I)

ELSE

INA(I) = INB(I-1) + 1

INB(I) = INB(I-1) + LONG(I)

END IF

TYPE(I) = VAL(RECDATA(13:14))

WHERE(I) = RECDATA(15:23)

4005 CONTINUE

K = 1

DO WHILE (INB(NFIELD)/10**K.GT.0)

K = K + 1

END DO

ENCODE(K,402,HOW(1)) LENREC

```

C      CALL DELCAT
C
C      ELSE
C
C      SELECT AND EXECUTE A CATEGORY REQUEST
C
C      CALL OPENCAT
3045  WRITE(22,317) CATNAME
      WRITE(22,318)
      READ(21,302) ANS
      CALL CHECK(ANS,ACT,TEN,TYPO)
      IF (TYPO) GO TO 3045
C
      IF (ACT.EQ.1) THEN
        CALL ADDREC
      ELSE IF (ACT.EQ.2) THEN
        CALL GETREC
      ELSE IF (ACT.EQ.3) THEN
        CALL DELREC
      ELSE IF (ACT.EQ.4) THEN
        CALL MODREC
      ELSE IF (ACT.EQ.5) THEN
        CALL QUERY
      ELSE IF (ACT.EQ.6) THEN
        CALL CATLIST
      ELSE IF (ACT.EQ.7) THEN
        CALL MODWORD
      ELSE IF (ACT.EQ.8) THEN
        CALL REVIEW
      ELSE IF (ACT.EQ.9) THEN
        READ(1,303,REC=1) I,J
        K = J - I
        WRITE(6,321) CATNAME,K
      ELSE IF (ACT.EQ.10) THEN
        DO 3060 I=1,NFIELD
          LDU = IO(I) + 10
          CLOSE(UNIT=LDU)
3060  CONTINUE
        RETURN
      END IF
      GO TO 3045
C
      END IF
      END

```

```

      KOUNT = 1
      FLD(1) = 'TABLE'
      WRITE(22,301) KOUNT,FLD(1)
      LTR = 'F'
3010 CALL BTREE(LTR,NINE,A,MAXLEN,IERR,IERR)
      IF (IERR.EQ.4.OR.IERR.EQ.5) GO TO 3020
      LTR = 'S'
      KOUNT = KOUNT + 1
      FLD(KOUNT) = A
      LINK(KOUNT) = IREC
      WRITE(22,301) KOUNT,A
      IF (MOD(KOUNT,20).EQ.0) THEN
3015   WRITE(22,311)
        READ(21,302) ANS
        CALL CHECK(ANS,N,KOUNT,TYP0)
        IF (TYP0) GO TO 3015
        IF (N.NE.0) GO TO 3030
        KOUNT = 0
      END IF
      GO TO 3010
3020 WRITE(22,312)
      WRITE(22,313)
      READ(21,302) ANS
      CALL CHECK(ANS,N,KOUNT,TYP0)
      IF (TYP0) GO TO 3020
      IF (N.EQ.0) RETURN
C
3030 IF (FLD(N)(1:5).EQ.'TABLE') THEN
      IF (NUM.EQ.1) THEN
        CALL TABMENU
      ELSE IF (NUM.EQ.2) THEN
        WRITE(22,314)
      END IF
      RETURN
    END IF
    READ(19,HOW(9),REC=LINK(N)) RECDATA
    CATNAME = RECDATA(1:9)
    CATWORD = RECDATA(10:19)
    NFIELD = VAL(RECDATA(20:21))
    KEYFLD = VAL(RECDATA(22:23))
    IF (CATWORD.EQ.' ') GO TO 3035
    WRITE(22,315)
    READ(21,302) TRY
    IF (TRY.EQ.CATWORD) THEN
      GO TO 3035
    ELSE
      WRITE(22,316)
      RETURN
    END IF
3035 IF (NUM.EQ.2) THEN
C
C   DELETE A CATEGORY

```

SUBROUTINE PICKCAT(NUM)

IMPLICIT INTEGER*4 (A-Z)

COMMON /XXXBOSS/

1 NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2 IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3 ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4 IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
BYTE INA,INB,EXA,EXB,IO,IERR,NREF
CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

INTEGER LINK(20)

CHARACTER TRY*10,CATWORD*10

LOGICAL*1 THERE,TYPO

301 FORMAT(11X,I3,15X,A10)

302 FORMAT(A10)

303 FORMAT(2I5)

310 FORMAT(/10X,'NUMBER',10X,'CATEGORY NAME'/)

311 FORMAT(/3X,'ENTER THE APPROPRIATE NUMBER OR ENTER'/

*7X,'ZERO TO SEE MORE LIST')

312 FORMAT(/3X,'ENTER THE APPROPRIATE NUMBER')

313 FORMAT(3X,'OR ENTER ZERO TO RETURN TO PREVIOUS MENU')

314 FORMAT(/3X,'REQUEST DENIED - "TABLE" CANNOT BE DELETED'/)

315 FORMAT(/3X,'ENTER CATEGORY PASSWORD')

316 FORMAT(/3X,'INCORRECT CATEGORY PASSWORD')

317 FORMAT(/15X,'THE CURRENT CATEGORY IS ',A8,//)

318 FORMAT(10X,'NUMBER',10X,'ACTION'//

*12X,'1',13X,'ADD DATA'/

*12X,'2',13X,'GET DATA'/

*12X,'3',13X,'DELETE DATA'/

*12X,'4',13X,'MODIFY DATA'/

*12X,'5',13X,'RANGE QUERY'/

*12X,'6',13X,'LIST ENTIRE CATEGORY'/

*12X,'7',13X,'CHANGE CATEGORY PASSWORD'/

*12X,'8',13X,'DISPLAY RECORD FORMAT'/

*12X,'9',13X,'CURRENT NUMBER OF RECORDS'/

*12X,'10',12X,'RETURN TO PREVIOUS MENU'/

*/3X,'ENTER APPROPRIATE NUMBER')

321 FORMAT(/3X,'CURRENT NUMBER OF RECORDS IN CATEGORY ',A9,' = ',I6)

C DRIVER FOR CATEGORY REQUEST
C -----

C
C SELECT A CATEGORY
C

3005 WRITE(22,310)

```

WRITE(22,612) K,B(J)
IF (J.LE.2) THEN
  K = 3
  WRITE(22,613) K, LONG(I)
ELSE
  LONG(I) = 2
  IF (J.EQ.5) THEN
    K = 4
    WRITE(22,614) K, WHERE(I)
  END IF
END IF
IF (I.EQ.KEYFLD) WRITE(22,615)
IF (.NOT.NEW) RETURN
6015 WRITE(22,616)
READ(21,601) ANS
CALL CHECK(ANS, NUM, FOUR, TYPO)
IF (TYPO) GO TO 6015
IF (NUM.EQ.0) GO TO 6020
WRITE(22,617)
IF (NUM.EQ.1) THEN
  READ(21,601) TITLE(I)
ELSE IF (NUM.EQ.2) THEN
  WRITE(22,618)
  READ(21,*) TYPE(I)
ELSE IF (NUM.EQ.3) THEN
  READ(21,*) LONG(I)
ELSE IF (NUM.EQ.4) THEN
  READ(21,601) WHERE(I)
END IF
GO TO 6010
6020 RECDATA(1:10) = TITLE(I)
RECDATA(11:12) = SYM(LONG(I))
RECDATA(13:14) = SYM(TYPE(I))
RECDATA(15:23) = WHERE(I)
WRITE(20, HOW(10), REC=I) RECDATA
RETURN
END

```

```

SUBROUTINE VIEWSPEC(I,NEW)
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  CHARACTER SYM*2,B(5)*10
  LOGICAL*1 NEW,TYPE0
C
  DATA B/'CHARACTER','NUMERICAL','DATE','TABLE','DUPLICATE'/
C
601 FORMAT(A10)
610 FORMAT(/15X,'FIELD',I3,'  PARAMETERS')
611 FORMAT(/5X,I3,5X,'TITLE',T50,A10)
612 FORMAT(5X,I3,5X,'DATA TYPE',T50,A10)
613 FORMAT(5X,I3,5X,'MAXIMUM NUMBER OF CHARACTERS',T50,I3)
614 FORMAT(5X,I3,5X,'RESOURCE CATEGORY',T50,A9)
615 FORMAT(/5X,'*** - KEY FIELD')
616 FORMAT(/3X,'ENTER ZERO IF SATISFACTORY OR ENTER THE NUMBER'/
  *8X,'OF THE PARAMETER TO BE MODIFIED')
617 FORMAT(/3X,'ENTER NEW PARAMETER')
618 FORMAT(/10X'DATA TYPE'//
  *15X,'1 - CHARACTER'/
  *15X,'2 - NUMERICAL'/
  *15X,'3 - DATE (FROM 1 JAN 1900 TO 1 JAN 2076)'/
  *15X,'4 - TABLE'/
  *15X,'5 - DUPLICATE'/)
C
C-----
C      REVIEW THE FIELD PARAMETERS FOR A CATEGORY
C      (EDITING PERMITTED UPON CATEGORY CREATION)
C-----
C
  WRITE(22,610) I
6010 J = TYPE(I)
  IF (J.LE.3) THEN
    WHERE(I) = CATNAME
  ELSE IF (J.EQ.4) THEN
    WHERE(I) = 'TABLE'
  END IF
  K = 1
  WRITE(22,611) K,TITLE(I)
  K = 2

```

```

WRITE(22,212)
READ(21,201) CATWORD
WRITE(22,213)
READ(21,*) NFIELD
LENREC = 0
DO 2010 I=1,NFIELD
  WRITE(22,214) I
  READ(21,201) TITLE(I)
  WRITE(22,215)
  READ(21,*) TYPE(I)
  LONG(I) = 2
  IF (TYPE(I).LE.2) THEN
    WRITE(22,216)
    IF (TYPE(I).EQ.2) WRITE(22,217)
    READ(21,*) LONG(I)
  ELSE IF (TYPE(I).EQ.5) THEN
    WRITE(22,218)
    READ(21,201) WHERE(I)
  END IF
  LENREC = LENREC + LONG(I)
2010 CONTINUE
WRITE(22,219)
READ(21,*) KEYFLD
NEW = .TRUE.
DO 2020 I=1,NFIELD
  CALL VIEWSPEC(I,NEW)
2020 CONTINUE
LTR = 'A'
A = CATNAME
CALL BTREE(LTR,NINE,A,MAXLEN,IREC,IERR)
RECDATA = CATNAME
RECDATA(10:19) = CATWORD
RECDATA(20:21) = SYM(NFIELD)
RECDATA(22:23) = SYM(KEYFLD)
WRITE(19,HOW(9),REC=IREC) RECDATA
LTR = 'C'
A = CATNAME
MAXLEN = LONG(KEYFLD)
CALL BTREE(LTR,ONE,A,MAXLEN,IREC,IERR)
CLOSE(UNIT=11)
OPEN(UNIT=11,FILE=CATNAME//'.DAR',STATUS='NEW',FORM='FORMATTED',
*  ACCESS='DIRECT',RECL=LENREC)
RETURN
END

```

SUBROUTINE NEWCAT

C IMPLICIT INTEGER*4 (A-Z)

C COMMON /XXXBOSS/

1 NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
 2 IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
 3 ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
 4 IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
 BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
 BYTE INA,INB,EXA,EXB,IO,IERR,NREF
 CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
 CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

C CHARACTER SYM*2,CATWORD*10
 C LOGICAL*1 THERE,NEW

201 FORMAT(A10)
 210 FORMAT(/3X,'ENTER NAME OF NEW CATEGORY (AT MOST 9 CHARACTERS)')
 211 FORMAT(/10X,'CATEGORY NAME ALREADY IN USE - CHOOSE ANOTHER')
 212 FORMAT(/3X,'ENTER CATEGORY PASSWORD (AT MOST 10 LETTERS) -'/
 *5X,'IF NONE SIMPLY PRESS THE "RETURN" KEY')
 213 FORMAT(/3X,' ENTER NUMBER OF FIELDS PER DATA RECORD')
 214 FORMAT(/15X,'PARAMETERS OF FIELD',I3//
 *3X,'ENTER FIELD NAME (AT MOST 10 CHARACTERS)')
 215 FORMAT(/10X'DATA TYPE'//
 *15X,'1 - CHARACTER'/
 *15X,'2 - NUMERICAL'/
 *15X,'3 - DATE (FROM 1 JAN 1900 TO 1 JAN 2076)'/
 *15X,'4 - TABLE'/
 *15X,'5 - DUPLICATE'/
 */3X,'ENTER NUMBER CORRESPONDING TO DATA TYPE')
 216 FORMAT(/3X,'ENTER MAXIMUM NUMBER OF CHARACTERS')
 217 FORMAT(3X,'COUNTING SIGNS AND DECIMAL POINTS')
 218 FORMAT(/3X,'ENTER RESOURCE CATEGORY')
 219 FORMAT(/3X,'ENTER NUMBER OF KEY FIELD')

C-----
 C CREATE A NEW CATEGORY
 C-----
 C

WRITE(22,210)
 READ(21,201) CATNAME
 INQUIRE(FILE=CATNAME//'.DAR',EXIST=THEORE)
 IF (THERE) THEN
 WRITE(22,211)
 RETURN
 END IF
 CLOSE(UNIT=20)
 OPEN(UNIT=20,FILE=CATNAME//'.LAR',STATUS='NEW',FORM='FORMATTED',
 * ACCESS='DIRECT',RECL=23)

```

*10X,'3',10X,'CREATE A NEW CATEGORY'/
*10X,'4',10X,'EXIT'/
*/3X,'ENTER APPROPRIATE NUMBER')

C
C-----
C      OPEN THE DATA BASE
C-----
C
      OPEN(UNIT=21,FILE='SYS$INPUT',STATUS='UNKNOWN')
      OPEN(UNIT=22,FILE='SYS$OUTPUT',STATUS='UNKNOWN')
C
      HOW(9) = '(A23)'
      HOW(10) = '(A23)'
      INQUIRE(FILE='CAT.KEY',EXIST=THEIR)
      IF (THEIR) THEN
        LTR = 'O'
      ELSE
        LTR = 'C'
      END IF
      A = 'CAT'
      MAXLEN = 9
      CALL BTREE(LTR,NINE,A,MAXLEN,IREF,IERR)
      CLOSE(UNIT=19)
      OPEN(UNIT=19,FILE='CAT.DAR',STATUS='UNKNOWN',FORM='FORMATTED',
*      ACCESS='DIRECT',RECL=23)
      A = 'TABLE'
      MAXLEN = 20
      CALL BTREE(LTR,TEN,A,MAXLEN,IREF,IERR)
1010 WRITE(22,110)
      READ(21,101) ANS
      CALL CHECK(ANS,NUM,FOUR,TYPO)
      IF (TYPO) GO TO 1010
      IF (NUM.EQ.1.OR.NUM.EQ.2) THEN
        CALL PICKCAT(NUM)
      ELSE IF (NUM.EQ.3) THEN
        CALL NEWCAT
      ELSE IF (NUM.EQ.4) THEN
        STOP
      END IF
      GO TO 1010
      END

```

PROGRAM BOSS

```

C
C*****
C      BOSS is an interactive relational database manager
C      which uses a B+ tree for storing and retrieving record
C      keys. This implementation can accomodate up to 65,535
C      data records in any one category.
C      Complete documentation for BOSS is contained in
C      "BOSS : A FORTRAN Code for a Relational Database
C      Manager" by Elliot Winston, NSWC TR 85-56. Associated
C      documentation can be found in "BTREE : A FORTRAN
C      Code for a B+ Tree" by Elliot Winston, NSWC TR 85-54.
C*****
C
C      LDU          FILE          LDU          FILE
C
C      1      (NREF=1).KEY      11      (NREF=1).DAR
C      2      (NREF=2).KEY      12      (NREF=2).DAR
C      :      :      :      :
C      :      :      :      :
C      8      (NREF=8).KEY      18      (NREF=8).DAR
C      9      CAT.KEY      19      CAT.DAR
C      10     TABLE.KEY      20      (***) .LAR
C      21     INPUT (KEYBOARD)  22      OUTPUT (SCREEN)
C      23     SCRATCH          24      SCRATCH
C
C      MAXIMUM NUMBER OF FIELDS PER RECORD = 20
C      MAXIMUM FIELD LENGTH = 100 BYTES
C      MAXIMUM NUMBER OF BYTES PER RECORD = 256
C      MAXIMUM NUMBER OF RELATED CATEGORIES = 8
C
C      IMPLICIT INTEGER*4 (A-Z)
C
C      COMMON /XXXBOSS/
1      NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF, LONG(20),
2      IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3      ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4      IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
      BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
      BYTE INA,INB,EXA,EXB,IO,IERR,NREF
      CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
      CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
C      LOGICAL*1 THERE,TYPE
C
C      DATA ONE,TWO,THREE,FOUR,FIVE/1,2,3,4,5/
C      DATA SIX,SEVEN,EIGHT,NINE,TEN/6,7,8,9,10/
C
101  FORMAT(A3)
110  FORMAT('0',9X,'1',10X,'ACCESS AN ACTIVE CATEGORY'/
      *10X,'2',10X,'DELETE AN ACTIVE CATEGORY'/

```

NSWC TR 85-56

APPENDIX B
FORTRAN CODE LISTING

FUNCTION CONVERT

PURPOSE: To convert a number in string format into its real numerical value.

INPUTS:

A CHARACTER*15 number in string format

OUTPUTS:

CONVERT REAL*4 real value associated with A

EXTERNALS:

none

FUNCTION DEDATE

PURPOSE: To convert the number of days since December 31,1899
into the format MONTH/DAY/YEAR .

INPUTS:

MANY INTEGER*4 number of days since December 31,1899

OUTPUTS:

DEDATE CHARACTER*10 date associated with MANY

EXTERNALS:

none

```

SUBROUTINE DELCAT
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  CHARACTER INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  CHARACTER NAME*9
C
301 FORMAT(A3)
310 FORMAT(/' REQUEST DENIED - ',A10,' IS REFERENCED BY ',A10)
311 FORMAT(/3X,'DO YOU WISH TO DELETE CATEGORY ',A9,'? (Y/N)')
312 FORMAT(/3X' REQUEST TO DELETE CATEGORY ',A9,' IS CANCELLED')
C
C-----
C      DRIVER TO DELETE A CATEGORY
C-----
C
C      CHECK FOR RELATED CATEGORIES
C
  CLOSE(UNIT=20)
  LTR = 'F'
3010 CALL BTREE(LTR,NINE,A,MAXLEN,IERR,IERR)
  IF (IERR.EQ.4.OR.IERR.EQ.5) GO TO 3030
  LTR = 'S'
  READ(19,HOW(9),REC=IERR) RECDATA
  NAME RECDATA(1:9)
  END = VAL(RECDATA(20:21))
  IF (NAME.EQ.CATNAME) GO TO 3010
  OPEN(UNIT=20,FILE=NAME//'.LAR',STATUS='OLD',FORM='FORMATTED',
*    ACCESS='DIRECT')
  DO 3020 I=1,END
    READ(20,HOW(10),REC=I) RECDATA
    WHERE(I) = RECDATA(15:23)
    IF (WHERE(I).EQ.CATNAME) THEN
      WRITE(22,310) CATNAME,NAME
      RETURN
    END IF
  3020 CONTINUE
  CLOSE(UNIT=20)
  GO TO 3010
C
C      DELETION OF CATEGORY
C

```

```
3030 CLOSE(UNIT=20)
      WRITE(22,311) CATNAME
      READ(21,301) ANS
      IF (ANS(1:1).EQ.'Y') THEN
        LTR = 'D'
        A = CATNAME
        CALL BTREE(LTR,NINE,A,MAXLEN,IREC,IERR)
        CLOSE(UNIT=1)
        OPEN(UNIT=1,FILE=A//'.KEY',STATUS='OLD',FORM='FORMATTED',
*          ACCESS='DIRECT')
        CLOSE(UNIT=1,STATUS='DELETE')
        CLOSE(UNIT=11)
        OPEN(UNIT=11,FILE=A//'.DAR',STATUS='OLD',FORM='FORMATTED',
*          ACCESS='DIRECT')
        CLOSE(UNIT=11,STATUS='DELETE')
        CLOSE(UNIT=20)
        OPEN(UNIT=20,FILE=A//'.LAR',STATUS='OLD',FORM='FORMATTED',
*          ACCESS='DIRECT')
        CLOSE(UNIT=20,STATUS='DELETE')
        OPEN(UNIT=1,FILE=A//'.NOD',STATUS='UNKNOWN')
        CLOSE(UNIT=1,STATUS='DELETE')
        OPEN(UNIT=1,FILE=A//'.REC',STATUS='UNKNOWN')
        CLOSE(UNIT=1,STATUS='DELETE')
      ELSE IF (ANS(1:1).EQ.'N') THEN
        WRITE(22,312) CATNAME
      ELSE
        GO TO 3030
      END IF
      RETURN
      END
```

```

SUBROUTINE ADDREC
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
601 FORMAT(A100)
610 FORMAT(/10X,'ENTER THE DATA FOR FIELD',I3,' (',A10,')')
611 FORMAT(/5X,'THE FORMAT FOR A DATE IS  MM/DD/YYYY'//
  *15X,'MM      =  INTEGER FROM 1 TO 12  (MONTH)'/
  *15X,'DD      =  INTEGER FROM 1 TO 31  (DAY)'/
  *15X,'YYYY    =  4 DIGITS WHICH SPECIFY THE YEAR'/
  */3X,'ENTER THE DATE')
612 FORMAT(13X,'(AT MOST ',I3,' CHARACTERS)')
613 FORMAT(/3X,'ERROR - DATA IS REQUIRED FOR FIELD',I3)
614 FORMAT(/3X,'REQUEST TO ADD DATA IS DENIED')
C
C-----
C      DRIVER FOR ADDING A DATA RECORD
C-----
C
DO 6020 I=1,NFIELD
6010  WRITE(22,610) I,TITLE(I)
      IF (TYPE(I).EQ.3) THEN
        WRITE(22,611)
        READ(21,601) FLD(I)
      ELSE IF (TYPE(I).EQ.4) THEN
        IND = 3
        CALL TABLIST(TITLE(I),FLD(I),IND)
        IF (FLD(I).EQ.' ') THEN
          WRITE(22,614)
          RETURN
        END IF
      ELSE
        K = EXB(I) - EXA(I) + 1
        WRITE(22,612) K
        READ(21,601) FLD(I)
      END IF
      IF ((I.EQ.KEYFLD.OR.TYPE(I).GE.3).AND.FLD(I).EQ.' ') THEN
        WRITE(6,613) I
        GO TO 6010
      END IF
6020 CONTINUE

```

```
CALL VERIFY  
CALL RECIN  
IF (IERR.EQ.4) THEN  
    WRITE(22,614)  
    RETURN  
END IF  
CALL INSERT  
RETURN  
END
```

```

SUBROUTINE VERIFY
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  LOGICAL*1 TYPO
C
  701 FORMAT(A100)
  710 FORMAT(/3X,'ENTER ZERO IF THE DATA IS CORRECT, OR ENTER THE '
    * /3X,'NUMBER OF THE FIELD WITH THE INCORRECT DATA')
  711 FORMAT(/3X,'ENTER THE CORRECT DATA')
C
C-----
C      VERIFY A DATA RECORD
C-----
C
  7010 CALL SHOWREC
  WRITE(22,710)
  READ(21,701) ANS
  CALL CHECK(ANS,N,NFIELD,TYPO)
  IF (TYPO) GO TO 7010
  IF (N.EQ.0) RETURN
  IF (TYPE(N).EQ.4) THEN
    IND = 3
    CALL TABLIST(TITLE(N),FLD(N),IND)
  ELSE
    WRITE(22,711)
    READ(21,701) FLD(N)
  END IF
  GO TO 7010
END

```

SUBROUTINE INSERT

```

C
C   IMPLICIT INTEGER*4 (A-Z)
C
C   COMMON /XXXBOSS/
1   NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2   IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3   ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4   IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
   BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
   BYTE INA,INB,EXA,EXB,IO,IERR,NREF
   CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
   CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
401 FORMAT(2I5)
410 FORMAT(/3X,'RECORD NOT INSERTED INTO THE DATABASE -'
   */3X,'THE KEY IS ALREADY IN USE')
411 FORMAT(/3X,'CATEGORY ',A9,' CONTAINS THE MAXIMUM NUMBER OF'
   */3X,'RECORDS ALLOWED - NO ADDITIONAL RECORDS WILL BE ADDED')
412 FORMAT(/3X,'WARNING - ',A9,' NOW CONTAINS',I6,' RECORDS;'
   */3X,'THE MAXIMUM NUMBER IS 65530')
C
C-----
C   INSERT A DATA RECORD INTO CURRENT CATEGORY
C-----
C
C   CHECK ON NUMBER OF CURRENT RECORDS
C
C   READ(1,401,REC=1) I,J
C   K = J - I
C   IF (K.GT.65530) THEN
C       WRITE(22,411) CATNAME
C       RETURN
C   END IF
C
C   LTR = 'A'
C   A = FLD(KEYFLD)
C   CALL BTREE(LTR,ONE,A,MAXLEN,IREF,IERR)
C   IF (IREF.GE.65475) WRITE(22,412) CATNAME,IREF
C   IF (IERR.EQ.6) THEN
C       WRITE(22,410)
C       RETURN
C   END IF
C   DO 4010 I=1,NFIELD
C       RECDATA(INA(I):INB(I)) = FLD(I)
4010 CONTINUE
C   WRITE(11,HOW(1),REC=IREF) RECDATA
C   RETURN
C   END

```

```

SUBROUTINE GETREC
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
601 FORMAT(A3)
610 FORMAT(/3X,'SEE NEXT RECORD OF KEY SEQUENCE? (Y/N)')
611 FORMAT(/3X,'THERE ARE NO MORE RECORDS IN ',A9)
C
C-----
C      DRIVER FOR GETTING A DATA RECORD
C-----
C
  CALL FETCH
  IF (IERR.EQ.4) RETURN
6010 CALL RECOUT
  CALL SHOWREC
6020 WRITE(22,610)
  READ(21,601) ANS
  IF (ANS(1:1).EQ.'Y') THEN
    LTR = 'S'
    CALL BTREE(LTR,ONE,A,MAXLEN,IERR,IERR)
    IF (IERR.EQ.5) THEN
      WRITE(22,611) CATNAME
      RETURN
    END IF
    READ(11,HOW(1),REC=IREC) RECDATA
    GO TO 6010
  ELSE IF (ANS(1:1).EQ.'N') THEN
    RETURN
  ELSE
    GO TO 6020
  END IF
  RETURN
END

```

SUBROUTINE FETCH

C IMPLICIT INTEGER*4 (A-Z)

C COMMON /XXXBOSS/

1 NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF, LONG(20),
 2 IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
 3 ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
 4 IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
 BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
 BYTE INA,INB,EXA,EXB,IO,IERR,NREF
 CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
 CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

C CHARACTER SYM*2,WHEN*10

C 701 FORMAT(A10)

710 FORMAT(/5X,'THE FORMAT FOR A DATE IS MM/DD/YYYY'//

*15X,'MM = INTEGER FROM 1 TO 12 (MONTH)'/

*15X,'DD = INTEGER FROM 1 TO 31 (DAY)'/

*15X,'YYYY = 4 DIGITS WHICH SPECIFY THE YEAR'/

*/3X,'ENTER THE DATE')

711 FORMAT(/3X,'ENTER THE VALUE OF ',A10)

712 FORMAT(/3X,'NO RECORD IN ',A9,' HAS THE REQUESTED KEY')

C-----
 C RETRIEVE A DATA RECORD FROM CURRENT CATEGORY
 C-----

C K = TYPE(KEYFLD)

IF (K.EQ.3) THEN

WRITE(22,710)

READ(21,701) WHEN

A = SYM(ENDATE(WHEN))

ELSE IF (K.EQ.4) THEN

IND = 3

CALL TABLIST(TITLE(KEYFLD),A,IND)

ELSE

WRITE(22,711) TITLE(KEYFLD)

READ(21,701) A

IF (K.EQ.5) THEN

LTR = 'G'

CALL BTREE(LTR,IO(KEYFLD),A,MAXLEN,IREF,IERR)

IF (IERR.EQ.4) THEN

WRITE(22,712) WHERE(KEYFLD)

RETURN

END IF

A = SYM(IREF)

END IF

END IF

LTR = 'G'

```
CALL BTREE(LTR,ONE,A,MAXLEN,IERR,IERR)
IF (IERR.EQ.4) THEN
  WRITE(22,712) CATNAME
ELSE
  READ(11,HOW(1),REC=IREC) RECDATA
END IF
RETURN
END
```

SUBROUTINE DELREC

IMPLICIT INTEGER*4 (A-Z)

COMMON /XXXBOSS/

```

1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

```

CHARACTER NAME*9,COPYCAT*9,UNIQUE*20

```

302 FORMAT(A10)
310 FORMAT(/3X,'ENTER FULL KEY VALUE OF ',A10)
311 FORMAT(/3X,'NO RECORD IN ',A9,' HAS THE REQUESTED KEY')
312 FORMAT(/3X,'RECORD DELETION REQUEST CANCELLED')
313 FORMAT(/3X,' REQUEST DENIED - REFERENCED IN A DATA RECORD'/
  *3X,'CONTAINED IN CATEGORY ',A9)
314 FORMAT(/3X,'DO YOU WISH TO DELETE THIS RECORD? (Y/N)')
315 FORMAT(/3X,'PRESS THE "RETURN" KEY TO CONTINUE')
316 FORMAT(/3X,'KEYSTROKE ERROR - TRY AGAIN')

```

 DRIVER FOR DELETING A DATA RECORD

```

  COPYCAT = CATNAME
  NF = NFIELD
  WRITE(22,310) TITLE(KEYFLD)
  READ(21,302) UNIQUE
  LTR = 'G'
  A = UNIQUE
  CALL BTREE(LTR,ONE,A,MAXLEN,KEYREC,IERR)
  IF (IERR.EQ.4) THEN
    WRITE(22,311) CATNAME
    WRITE(22,312)
    RETURN
  END IF

```

CHECK FOR RELATED RECORDS

```

  LTR = 'F'
3010 CALL BTREE(LTR,NINE,A,MAXLEN,IERR,IERR)
  IF (IERR.EQ.5) GO TO 3040
  LTR = 'S'
  READ(19,HOW(9),REC=IERR) RECDATA
  CATNAME = RECDATA(1:9)

```

```

NFIELD = VAL(RECDATA(20:21))
IF (CATNAME.EQ.COPYCAT) GO TO 3010
CLOSE(UNIT=20)
OPEN(UNIT=20,FILE=CATNAME//'.LAR',STATUS='OLD',
*   FORM='FORMATTED',ACCESS='DIRECT')
DO 3030 I=1,NFIELD
  READ(20,HOW(10),REC=I) RECDATA
  IF (RECDATA(15:23).EQ.COPYCAT) THEN
    CALL OPENCAT
    LTR = 'F'
3020   CALL BTREE(LTR,ONE,A,MAXLEN,IERR,IERR)
    IF (IERR.EQ.4.OR.IERR.EQ.5) GO TO 3030
    LTR = 'S'
    READ(11,HOW(1),REC=IERR) RECDATA
    K = VAL(RECDATA(INA(I):INB(I)))
    IF (K.EQ.KEYREC) THEN
      WRITE(22,313) CATNAME
      CALL RECOUT
      CALL SHOWREC
      WRITE(22,315)
      READ(21,302) ANS
      CATNAME = COPYCAT
      NFIELD = NF
      CALL OPENCAT
      RETURN
    END IF
    GO TO 3020
  END IF
3030 CONTINUE
  LTR = 'S'
  GO TO 3010

C
C   DELETION OF RECORD
C
3040 CATNAME = COPYCAT
  NFIELD = NF
  CALL OPENCAT
  READ(11,HOW(1),REC=KEYREC) RECDATA
  CALL RECOUT
  CALL SHOWREC
3050 WRITE(22,314)
  READ(21,302) ANS
  IF (ANS(1:1).EQ.'Y') THEN
    CALL RECIN
    LTR = 'D'
    A = FLD(KEYFLD)
    CALL BTREE(LTR,ONE,A,MAXLEN,IERR,IERR)
  ELSE IF (ANS(1:1).EQ.'N') THEN
    WRITE(22,312)
  ELSE
    WRITE(22,316)
    GO TO 3050
  
```

NSWC TR 85-56

END IF
RETURN
END

SUBROUTINE MODREC

C

IMPLICIT INTEGER*4 (A-Z)

C

COMMON /XXXBOSS/

```
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
   BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
   BYTE INA,INB,EXA,EXB,IO,IERR,NREF
   CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
   CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
```

C

CHARACTER NAME*9,OLDKEY*20
LOGICAL*1 NOTE

C

```
401 FORMAT(A3)
410 FORMAT(/3X,'APPROPRIATE CHANGES IN ',A9,' WILL BE MADE')
411 FORMAT(/3X,'DO YOU WISH TO MAKE THE MODIFICATION? (Y/N)')
412 FORMAT(/3X,'REQUEST TO MODIFY DATA DENIED')
413 FORMAT(/3X,'REQUEST TO MODIFY DATA CANCELLED')
```

C

C

C

C

C

```
-----
      DRIVER FOR MODIFYING A DATA RECORD
      -----
NOTE = .FALSE.
CALL FETCH
IF (IERR.EQ.4) RETURN
OLDKEY = RECDATA(INA(KEYFLD):INB(KEYFLD))
CALL RECOUT
CALL VERIFY
```

C

C

C

CHECK FOR CATEGORIES AFFECTED BY THE MODIFICATION

```
LTR = 'F'
4010 CALL BTREE(LTR,NINE,A,MAXLEN,IREC,IERR)
IF (IERR.EQ.5) GO TO 4030
LTR = 'S'
READ(19,HOW(9),REC=IREC) RECDATA
NAME = RECDATA(1:9)
END = VAL(RECDATA(20:21))
IF (NAME.EQ.CATNAME) GO TO 4010
CLOSE(UNIT=20)
OPEN(UNIT=20,FILE=NAME//'.LAR',STATUS='OLD',FORM='FORMATTED',
*  ACCESS='DIRECT')
DO 4020 I=1,END
  READ(20,HOW(10),REC=I) RECDATA
  WHERE(I) = RECDATA(15:23)
  IF (WHERE(I).EQ.CATNAME) THEN
```

```

        WRITE(22,410) NAME
        NOTE = .TRUE.
        GO TO 4010
    END IF
4020 CONTINUE
    GO TO 4010
C
C      MODIFICATION OF RECORD
C
4030 CLOSE(UNIT=20)
    IF (NOTE) THEN
        WRITE(22,411)
        READ(21,401) ANS
    ELSE
        ANS = 'Y'
    END IF
    IF (ANS(1:1).EQ.'Y') THEN
        CALL RECIN
        IF (IERR.EQ.4) THEN
            WRITE(22,412)
            RETURN
        END IF
        LTR = 'D'
        CALL BTREE(LTR,ONE,OLDKEY,MAXLEN,IERR,IERR)
        CALL INSERT
    ELSE IF (ANS(1:1).EQ.'N') THEN
        WRITE(22,413)
    ELSE
        GO TO 4030
    END IF
    RETURN
END

```

SUBROUTINE TABMENU

IMPLICIT INTEGER*4 (A-Z)

COMMON /XXXBOSS/

```

1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

```

```

  CHARACTER FLDNAME*10,FLDVAL*10
  LOGICAL*1 TYPO

```

```

901 FORMAT(A3)
910 FORMAT(/15X,'FIELD NAMES IN "TABLE"'/
  */10X,'NUMBER',10X,'NAME'//)
911 FORMAT(11X,I3,12X,A10)
912 FORMAT(/3X,'ENTER THE APPROPRIATE NUMBER')
913 FORMAT(3X,'OR ENTER ZERO TO SEE MORE LIST')
914 FORMAT(3X,'OR ENTER ZERO TO RETURN TO PREVIOUS MENU')
915 FORMAT(/10X,'NUMBER',10X,'ACTION'//
  *12X,'1',13X,'MODIFY DATA'//
  *12X,'2',13X,'DELETE DATA'//
  *12X,'3',13X,'LIST CURRENT FIELD VALUES'//
  *12X,'4',13X,'ADD DATA'//
  *12X,'5',13X,'RETURN TO PREVIOUS MENU')

```

```

-----
  DRIVER FOR "TABLE" REQUEST
-----

```

LIST ALL "TABLE" FIELDS OF ALL CATEGORIES

```

900 WRITE(22,910)
  KOUNT = 0
  LTR = 'F'
905 CALL BTREE(LTR,NINE,A,MAXLEN,IREF,IERR)
  IF (IERR.EQ.4.OR.IERR.EQ.5) GO TO 9020
  LTR = 'S'
  READ(19,HOW(9),REC=IREF) RECDATA
  CATNAME = RECDATA(1:9)
  NFIELD = VAL(RECDATA(20:21))
  CLOSE(UNIT=20)
  OPEN(UNIT=20,FILE=CATNAME//'.LAR',STATUS='OLD',
  *   FORM='FORMATTED',ACCESS='DIRECT')
  DO 9010 I=1,NFIELD
    READ(20,HOW(10),REC=I) RECDATA

```

```
      M = N + 1
      N = N + WIDTH(J)
      KM = 16
      KN = KM + WIDTH(J)
      DO 5065 K=1,NCOL
        LINE(KM:KN) = COL(K)(M:N)
        KM = KM + TAB
        KN = KM + WIDTH(J)
5065    CONTINUE
        WRITE(24,514) LINE
5070 CONTINUE
      LINE = ' '
      WRITE(24,514) LINE
      IF (IND.EQ.0) GO TO 5040
      CLOSE(UNIT=23)
      CLOSE(UNIT=24)
      RETURN
      END
```

```

ELSE
  MUCH = 0
0020  MUCH = MUCH + 1
      WRITE(22,512)
      CALL FLDLIST(LINK(MUCH))
      WRITE(22,513)
      READ(21,502) ANS
      IF (ANS(1:1).EQ.'Y') GO TO 5020
END IF

      COMPUTE FORMAT PARAMETERS OF OUTPUT FILE

DO 5030 I=1,MUCH
  J = LINK(I)
  IF (TYPE(J).EQ.3.OR.TYPE(J).EQ.4) THEN
    WIDTH(J) = 10
  ELSE
    WIDTH(J) = EXB(J) - EXA(J) + 1
  END IF
0030 CONTINUE
  TAB = 0
  DO 5035 I=1,MUCH
    TAB = MAX0(TAB,WIDTH(LINK(I)))
0035 CONTINUE
  TAB = TAB + 5
  NCOL = MIN0(115/TAB,8)

      WRITE SELECTED RECORD FIELDS

      REWIND(UNIT=23)
      OPEN(UNIT=24,FILE=CATNAME//'.OUT',STATUS='NEW')
0040 DO 5045 K=1,NCOL
      COL(K) = ' '
0045 CONTINUE
      DO 5055 K=1,NCOL
        READ(23,501,END=5060,IOSTAT=IND) IREC
        READ(11,HOW(1),REC=IREC) RECDATA
        CALL RECOUT
        M = 0
        N = 0
        DO 5050 I=1,MUCH
          J = LINK(I)
          M = N + 1
          N = N + WIDTH(J)
          COL(K)(M:N) = FLD(J)
0050 CONTINUE
0055 CONTINUE
0060 M = 0
      N = 0
      DO 5070 I=1,MUCH
        J = LINK(I)
        LINE = TITLE(J)

```

SUBROUTINE OUTPUT

C IMPLICIT INTEGER*4 (A-Z)

C COMMON /XXXBOSS/

1 NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
 2 IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
 3 ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
 4 IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
 BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
 BYTE INA,INB,EXA,EXB,IO,IERR,NREF
 CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
 CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

C INTEGER WIDTH(20)

BYTE LINK(20)

CHARACTER LINE*132,COL(8)*256

C 501 FORMAT(I5)

502 FORMAT(A3)

510 FORMAT(/3X,'DO YOU WISH TO CREATE THE OUTPUT FILE ',A12
 */3X,' CONTAINING ALL THE RECORDS FOUND? (Y/N)')

511 FORMAT(/3X,'DO YOU WISH TO WRITE ALL FIELDS? (Y/N)')

512 FORMAT(/3X,'SELECT A FIELD TO BE WRITTEN'/)

513 FORMAT(/3X,'DO YOU WISH TO WRITE AN ADDITIONAL FIELD? (Y/N)')

514 FORMAT(' ',A132)

C -----
 C WRITE A SET OF RECORDS ON AN OUTPUT FILE
 C -----

C SELECT RECORD FIELDS TO BE WRITTEN

C J = 9

DO WHILE (CATNAME(J:J).EQ.' ')

J = J - 1

END DO

A = CATNAME(1:J)//'.OUT'

WRITE(22,510) A

READ(21,502) ANS

IF (ANS(1:1).EQ.'N') THEN

CLOSE(UNIT=23)

RETURN

END IF

WRITE(22,511)

READ(21,502) ANS

IF (ANS(1:1).EQ.'Y') THEN

MUCH = NFIELD

DO 5015 I=1,MUCH

LINK(I) = I

5015 CONTINUE

SUBROUTINE CATLIST

IMPLICIT INTEGER*4 (A-Z)

COMMON /XXXBOSS/

```

1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

```

701 FORMAT(I5)

710 FORMAT(/3X,I5,' RECORDS WERE FOUND')

```

-----
      WRITE NUMBER OF EVERY RECORD OF CURRENT
      CATEGORY ON A SCRATCH FILE
-----

```

OPEN(UNIT=23,FILE='RECNOS',STATUS='SCRATCH')

KOUNT = 0

LTR = 'F'

7010 CALL BTREE(LTR,ONE,A,MAXLEN,IREF,IERR)

IF (IERR.EQ.4.OR.IERR.EQ.5) GO TO 7020

KOUNT = KOUNT + 1

LTR = 'S'

WRITE(23,701) IREF

GO TO 7010

7020 WRITE(22,710) KOUNT

CALL OUTPUT

RETURN

END

```

        SO = A.GE.BD(K)
      END IF
    ELSE
      IF (TYPE(J).EQ.3) THEN
        X = ENDATE(A)
        Y = VAL(BD(K))
      ELSE IF (TYPE(J).EQ.2) THEN
        X = CONVERT(A)
        Y = CONVERT(BD(K))
      END IF
      IF (L.EQ.1) THEN
        SO = X.EQ.Y
      ELSE IF (L.EQ.2) THEN
        SO = X.NE.Y
      ELSE IF (L.EQ.3) THEN
        SO = X.LT.Y
      ELSE IF (L.EQ.4) THEN
        SO = X.GT.Y
      ELSE IF (L.EQ.5) THEN
        SO = X.LE.Y
      ELSE IF (L.EQ.6) THEN
        SO = X.GE.Y
      END IF
    END IF
    IF (.NOT.SO) GO TO 8030
8045 CONTINUE
    MANY = MANY + 1
    WRITE(23,802) IREC
    GO TO 8030
C
C      OPTION TO WRITE ADMISSIBLE RECORDS TO A FILE
C
8050 WRITE(22,815) MANY
    CALL OUTPUT
    RETURN
  END

```

```

      CALL FLDLIST(I)
      LIST(KOUNT) = I
8015  WRITE(22,811)
      READ(21,801) ANS
      CALL CHECK(ANS, LINK(KOUNT), SIX, TYPO)
      IF (TYPO) GO TO 8015
      J = TYPE(I)
      IF (J.EQ.3) THEN
        WRITE(22,812)
        READ(21,801) WHEN
        BD(KOUNT) = SYM(ENDATE(WHEN))
      ELSE
        WRITE(22,813)
        READ(21,801) BD(KOUNT)
      END IF
8020  WRITE(22,814)
      READ(21,801) ANS
      IF (ANS(1:1).EQ.'Y') THEN
        GO TO 8010
      ELSE IF (ANS(1:1).EQ.'N') THEN
        GO TO 8025
      ELSE
        WRITE(22,816)
        GO TO 8020
      END IF
C
C      EXAMINE EVERY RECORD IN CATEGORY
C
8025  OPEN(UNIT=23, FILE='RECNO$', STATUS='SCRATCH')
      MANY = 0
      LTR = 'F'
8030  CALL BTREE(LTR, ONE, A, MAXLEN, IREC, IERR)
      IF (IERR.EQ.4.OR.IERR.EQ.5) GO TO 8050
      LTR = 'S'
      READ(11, HOW(1), REC=IREC) RECDATA
      CALL RECOUT
      DO 8045 K=1, KOUNT
        L = LINK(K)
        J = LIST(K)
        A = FLD(J)
        IF (TYPE(J).EQ.1.OR.TYPE(J).GE.4) THEN
          IF (L.EQ.1) THEN
            SO = A.EQ.BD(K)
          ELSE IF (L.EQ.2) THEN
            SO = A.NE.BD(K)
          ELSE IF (L.EQ.3) THEN
            SO = A.LT.BD(K)
          ELSE IF (L.EQ.4) THEN
            SO = A.GT.BD(K)
          ELSE IF (L.EQ.5) THEN
            SO = A.LE.BD(K)
          ELSE IF (L.EQ.6) THEN

```

SUBROUTINE QUERY

C

IMPLICIT INTEGER*4 (A-Z)

C

COMMON /XXXBOSS/

```

1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256

```

C

BYTE LINK(20),LIST(20)

REAL X,Y

CHARACTER SYM*2,WHEN*10,BD(20)*10

LOGICAL*1 SO,TYPO

C

801 FORMAT(A20)

802 FORMAT(I5)

810 FORMAT(/10X,'SELECT QUERY FIELD')

811 FORMAT(/10X,'NUMBER',10X,'RELATION'//

*13X,'1',27X,'EQUAL'//

*13X,'2',27X,'NOT EQUAL'//

*13X,'3',27X,'STRICTLY LESS THAN'//

*13X,'4',27X,'STRICTLY GREATER THAN'//

*13X,'5',27X,'LESS THAN OR EQUAL'//

*13X,'6',27X,'GREATER THAN OR EQUAL'//

*/3X,'ENTER APPROPRIATE NUMBER')

812 FORMAT(/5X,'THE FORMAT FOR A DATE IS MM/DD/YYYY'//

*15X,'MM = INTEGER FROM 1 TO 12 (MONTH)'/

*15X,'DD = INTEGER FROM 1 TO 31 (DAY)'/

*15X,'YYYY = 4 DIGITS WHICH SPECIFY THE YEAR'//

*/3X,'ENTER THE DATE')

813 FORMAT(/3X,'ENTER THE BOUND (NO MORE THAN 10 CHARACTERS)')

814 FORMAT(/3X,'DO YOU WISH TO SPECIFY MORE RELATIONS? (Y/N)')

815 FORMAT(/3X,I6,' RECORDS WERE FOUND')

816 FORMAT(/3X,'KEYSTROKE ERROR - TRY AGAIN')

C

C-----

C WRITE NUMBERS OF ALL DATA RECORDS SATISFYING A

C SET OF SPECIFIED CONDITIONS ON A SCRATCH FILE

C-----

C

C SELECT QUERY FIELDS AND SPECIFY CONDITIONS

C

KOUNT = 0

8010 KOUNT = KOUNT + 1

IF (KOUNT.EQ.20) GO TO 8025

WRITE(22,810)

```

SUBROUTINE SHOWREC
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
801 FORMAT(A3)
810 FORMAT(/)
811 FORMAT(3X,'FIELD',I3,3X,A10,5X,A100)
812 FORMAT(/3X,'DO YOU WISH TO SEE MORE LIST? (Y/N)')
C
C-----
C      DISPLAY A RECORD ON THE SCREEN
C-----
C
  WRITE(22,810)
  DO 8010 I=1,NFIELD
    WRITE(22,811) I,TITLE(I),FLD(I)
    IF (MOD(I,20).EQ.0) THEN
      WRITE(22,812)
      READ(21,801) ANS
      IF (ANS(1:1).EQ.'N') RETURN
    END IF
  8010 CONTINUE
  RETURN
END

```

```

SUBROUTINE RECOUT
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  CHARACTER DEDATE*10
C-----
C      TRANSFORM A RECORD FROM INTERNAL TO EXTERNAL FORMAT
C-----
C
  DO 4010 I=1,NFIELD
    FLD(I) = RECDATA(INA(I):INB(I))
4010 CONTINUE
  DO 4030 I=1,NFIELD
    IF (TYPE(I).LE.2) GO TO 4030
    K = VAL(FLD(I))
    IF (TYPE(I).EQ.3) THEN
      FLD(I) = DEDATE(K)
    ELSE IF (TYPE(I).EQ.4) THEN
      LTR = 'F'
      A = TITLE(I)
4020   CALL BTREE(LTR,TEN,A,MAXLEN,IREF,IERR)
      LTR = 'S'
      IF (IREF.NE.K) GO TO 4020
      FLD(I) = A(11:20)
    ELSE IF (TYPE(I).EQ.5) THEN
      LDU = IO(I) + 10
      READ(LDU,HOW(IO(I)),REC=K) RECDATA
      FLD(I) = RECDATA(EXA(I):EXB(I))
    END IF
4030 CONTINUE
  RETURN
  END

```

```

SUBROUTINE RECIN
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  CHARACTER SYM*2
C
  310 FORMAT(/3X,'KEY CANNOT BE FOUND IN ',A9)
C
C-----
C      TRANSFORM A RECORD FROM EXTERNAL TO INTERNAL FORMAT
C-----
C
  IERR = 0
  DO 3010 I=1,NFIELD
    IF (TYPE(I).EQ.3) THEN
      FLD(I) = SYM(ENDATE(FLD(I)))
    ELSE IF (TYPE(I).EQ.4) THEN
      LTR = 'G'
      A = TITLE(I)//FLD(I)
      CALL BTREE(LTR,TEN,A,MAXLEN,IERR,IERR)
      FLD(I) = SYM(IERR)
    ELSE IF (TYPE(I).EQ.5) THEN
      LTR = 'G'
      A = FLD(I)
      CALL BTREE(LTR,IO(I),A,MAXLEN,IERR,IERR)
      FLD(I) = SYM(IERR)
    END IF
    IF (IERR.EQ.4) THEN
      WRITE(22,310) WHERE(I)
      RETURN
    END IF
  3010 CONTINUE
  RETURN
END

```

```

SUBROUTINE FLDLIST(N)
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  LOGICAL*1 TYPO
C
  501 FORMAT(A3)
  510 FORMAT(/8X,'LIST OF FIELDS OF ',A9/)
  511 FORMAT(5X,'FIELD',I3,10X,A10)
  512 FORMAT(/3X,'ENTER THE APPROPRIATE FIELD NUMBER')
C
C-----
C      LIST FIELD NAMES OF CURRENT CATEGORY
C      AND SELECT A FIELD
C-----
C
  5010 WRITE(22,510) CATNAME
  DO 5020 I=1,NFIELD
    WRITE(22,511) I,TITLE(I)
  5020 CONTINUE
  WRITE(22,512)
  READ(21,501) ANS
  CALL CHECK(ANS,N,NFIELD,TYPO)
  IF (TYPO) GO TO 5010
  RETURN
  END

```

```

SUBROUTINE REVIEW
C
IMPLICIT INTEGER*4 (A-Z)
C
COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  LOGICAL*1 NEW
C
501 FORMAT(A3)
510 FORMAT(/3X,'SEE PARAMETERS OF ANOTHER FIELD? (Y/N)')
C
C-----
C      SELECT A FIELD AND REVIEW ITS PARAMETERS
C-----
C
  NEW = .FALSE.
5010 CALL FLDLIST(N)
  CALL VIEWSPEC(N,NEW)
  WRITE(22,510)
  READ(21,501) ANS
  IF (ANS(1:1).EQ.'Y') GO TO 5010
  RETURN
END

```

SUBROUTINE MODWORD

C

IMPLICIT INTEGER*4 (A-Z)

C

COMMON /XXXBOSS/

```
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
```

C

601 FORMAT(A10)

610 FORMAT(/3X,'ENTER NEW CATEGORY PASSWORD')

C

C

C

MODIFY PASSWORD OF CURRENT CATEGORY

C

C

LTR = 'G'

A = CATNAME

CALL BTREE(LTR,NINE,A,MAXLEN,IREF,IERR)

READ(19,HOW(9),REC=IREF) RECDATA

WRITE(22,610)

READ(21,601) RECDATA(10:19)

WRITE(19,HOW(9),REC=IREF) RECDATA

RETURN

END

```

TYPE(I) = VAL(RECDATA(13:14))
IF (TYPE(I).EQ.4) THEN
    KOUNT = KOUNT + 1
    FLD(KOUNT) = RECDATA(1:10)
    WRITE(22,911) KOUNT,FLD(KOUNT)
    IF (MOD(KOUNT,20).EQ.0) THEN
9008      WRITE(22,912)
          WRITE(22,913)
          READ(21,901) ANS
          CALL CHECK(ANS,N,KOUNT,TYPO)
          IF (TYPO) GO TO 9008
          IF (N.EQ.0) THEN
              KOUNT = 0
              WRITE(22,910)
          ELSE
              GO TO 9025
          END IF
    END IF
  END IF
9010 CONTINUE
    GO TO 9005
C
C      SELECT AND EXECUTE "TABLE" REQUEST
C
9020 WRITE(22,912)
    WRITE(22,914)
    READ(21,901) ANS
    CALL CHECK(ANS,N,KOUNT,TYPO)
    IF (TYPO) GO TO 9020
    IF (N.EQ.0) RETURN
9025 FLDNAME = FLD(N)
9030 WRITE(22,915)
    WRITE(22,912)
    READ(21,901) ANS
    CALL CHECK(ANS,NUM,FIVE,TYPO)
    IF (TYPO) GO TO 9030
    IF (NUM.LE.3) THEN
9040      CALL TABLIST(FLDNAME,FLDVAL,NUM)
          IF (NUM.LE.2) THEN
              IF (FLDVAL.EQ.' ') GO TO 9030
              CALL TABDEL(FLDNAME,FLDVAL,NUM)
              IF (NUM.EQ.1) THEN
                  CALL TABADD(FLDNAME)
                  GO TO 9040
              END IF
          END IF
          ELSE IF (NUM.EQ.4) THEN
              CALL TABADD(FLDNAME)
          ELSE IF (NUM.EQ.5) THEN
              GO TO 9000
          END IF
    GO TO 9030

```

END

```

SUBROUTINE TABADD(FLDNAME)
C
  IMPLICIT INTEGER*4 (A-Z)
C
  COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  CHARACTER FLDNAME*10,FLDVAL*10
C
  801 FORMAT(A10)
  810 FORMAT(/3X,'ENTER FIELD VALUE'/
    *3X,'(AT MOST 10 CHARACTERS)')
C
C-----
C      ADD A RECORD TO CATEGORY "TABLE"
C-----
C
  WRITE(22,810)
  READ(21,801) FLDVAL
  A(1:10) = FLDNAME
  A(11:20) = FLDVAL
  LTR = 'A'
  CALL BTREE(LTR,TEN,A,MAXLEN,IERR,IERR)
  RETURN
  END

```

```

SUBROUTINE TABDEL(FLDNAME,FLDVAL,NUM)
C
C   IMPLICIT INTEGER*4 (A-Z)
C
COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  CHARACTER FLDNAME*10,FLDVAL*10
C
401 FORMAT(A3)
410 FORMAT(/' REQUEST DENIED - REFERENCED BY CATEGORY ',A9)
411 FORMAT(/3X,'PRESS THE "RETURN" KEY TO CONTINUE')
412 FORMAT(/3X,'DO YOU WISH TO DELETE THE FIELD VALUE ',A10/
  *3X,'FROM THE FIELD NAMED ',A10,'? (Y/N)')
413 FORMAT(/3X,'DELETION REQUEST CANCELLED')
C
C-----
C      DRIVER FOR DELETING A "TABLE" FIELD VALUE
C-----
C
  A(1:10) = FLDNAME
  A(11:20) = FLDVAL
  LTR = 'G'
  CALL BTREE(LTR,TEN,A,MAXLEN,KEYREC,IERR)
C
C      CHECK FOR RELATED RECORDS
C
  LTR = 'F'
4005 CALL BTREE(LTR,NINE,A,MAXLEN,IREF,IERR)
  IF (IERR.EQ.4.OR.IERR.EQ.5) GO TO 4035
  LTR = 'S'
  READ(19,HOW(9),REC=IREF) RECDATA
  CATNAME = RECDATA(1:9)
  NFIELD = VAL(RECDATA(20:21))
  CLOSE(UNIT=20)
  OPEN(UNIT=20,FILE=CATNAME//'.LAR',STATUS='OLD',
  *   FORM='FORMATTED',ACCESS='DIRECT')
  DO 4025 I=1,NFIELD
    READ(20,HOW(10),REC=I) RECDATA
    TITLE(I) = RECDATA(1:10)
    TYPE(I) = VAL(RECDATA(13:14))
    IF (TITLE(I).EQ.FLDNAME.AND.TYPE(I).EQ.4) THEN
      CALL OPENCAT
      LTR = 'F'

```

```

4010      CALL BTREE(LTR,ONE,A,MAXLEN,IERR,IERR)
         IF (IERR.EQ.4.OR.IERR.EQ.5) THEN
             DO 4020 N=1,NREF
                 K = N + 10
                 CLOSE(UNIT=K)
4020      CONTINUE
             GO TO 4025
         END IF
         READ(11,HOW(1),REC=IREC) RECDATA
         K = VAL(RECDATA(INA(I):INB(I)))
         IF (K.EQ.KEYREC) THEN
             WRITE(22,410) CATNAME
             CALL RECOUT
             CALL SHOWREC
             WRITE(22,411)
             READ(21,401) ANS
             RETURN
         END IF
         LTR = 'S'
         GO TO 4010
     END IF
4025 CONTINUE
     LTR = 'S'
     GO TO 4005
C
C      DELETE A "TABLE" FIELD VALUE
C
4035 IF (NUM.EQ.2) THEN
     WRITE(22,412) FLDVAL,FLDNAME
     READ(21,401) ANS
     IF (ANS(1:1).EQ.'Y') THEN
         GO TO 4040
     ELSE IF (ANS(1:1).EQ.'N') THEN
         WRITE(22,413)
         RETURN
     ELSE
         GO TO 4035
     END IF
END IF
C
4040 LTR = 'D'
     A(1:10) = FLDNAME
     A(11:20) = FLDVAL
     CALL BTREE(LTR,TEN,A,MAXLEN,IERR,IERR)
     RETURN
END

```

```

SUBROUTINE TABLIST(FLDNAME,FLDVAL,IND)
C
IMPLICIT INTEGER*4 (A-Z)
C
COMMON /XXXBOSS/
1  NFIELD,KEYFLD,CATNAME,RECDATA,A,NREF,LONG(20),
2  IO(20),ANS,TYPE(20),FLD(20),TITLE(20),WHERE(20),
3  ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,
4  IERR,INA(20),INB(20),EXA(20),EXB(20),HOW(10),LTR
  BYTE ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN
  BYTE INA,INB,EXA,EXB,IO,IERR,NREF
  CHARACTER LTR*1,HOW*6,CATNAME*9,WHERE*9,TITLE*10
  CHARACTER ANS*3,A*20,FLD*100,RECDATA*256
C
  CHARACTER FLDNAME*10,FLDVAL*10,TEMP(20)*10
  LOGICAL*1 TYPO
C
201 FORMAT(A3)
210 FORMAT(/15X,'FIELD VALUES FOR ',A10//
  *10X,'NUMBER',10X,'FIELD VALUE'/)
211 FORMAT(11X,I3,2X,A20)
212 FORMAT(/)
213 FORMAT(3X,'ENTER THE APPROPRIATE NUMBER OR')
214 FORMAT(3X,'ENTER ZERO TO SEE MORE LIST')
215 FORMAT(3X,'ENTER THE APPROPRIATE NUMBER')
216 FORMAT(3X,'ENTER ZERO IF SATISFACTORY')
217 FORMAT(3X,'ENTER ZERO TO CONTINUE')
C
C-----
C      LIST ALL "TABLE" VALUES FOR A GIVEN FIELD
C-----
C
  WRITE(22,210) FLDNAME
  KOUNT = 0
  LTR = 'G'
  A = FLDNAME
2010 CALL BTREE(LTR,TEN,A,MAXLEN,IREC,IERR)
  LTR = 'S'
  IF (IERR.EQ.4.OR.IERR.EQ.5) GO TO 2020
  IF (A(1:10).EQ.FLDNAME) THEN
    KOUNT = KOUNT + 1
    TEMP(KOUNT) = A(11:20)
    WRITE(22,211) KOUNT,TEMP(KOUNT)
    IF (MOD(KOUNT,20).NE.0) GO TO 2010
2015  WRITE(22,212)
    IF (IND.LE.2) WRITE(22,213)
    WRITE(22,214)
    READ(21,201) ANS
    CALL CHECK(ANS,N,KOUNT,TYPO)
    IF (TYPO) GO TO 2015
    IF (N.EQ.0) THEN

```

```
        KOUNT = 0
        GO TO 2010
    ELSE
        GO TO 2030
    END IF
END IF
GO TO 2010
C
2020 WRITE(22,212)
    IF (IND.EQ.1) THEN
        WRITE(22,213)
        WRITE(22,216)
    ELSE IF (IND.EQ.2) THEN
        WRITE(22,215)
    ELSE IF (IND.EQ.3) THEN
        WRITE(22,217)
    END IF
    READ(21,201) ANS
    CALL CHECK(ANS,N,KOUNT,TYP0)
    IF (TYP0) GO TO 2020
    IF (N.EQ.0) THEN
        FLDVAL = ' '
        RETURN
    END IF
C
2030 FLDVAL = TEMP(N)
    RETURN
END
```

```

SUBROUTINE CHECK(ANS,N,NMAX,TYPO)
C
  CHARACTER ANS*3,B*1
  LOGICAL*1 TYPO
C
  401 FORMAT(I<L>)
  410 FORMAT(/3X,'KEYSTROKE ERROR - TRY AGAIN')
C
C-----
C      TRAP FOR A PARTICULAR CLASS OF TYPOGRAPHICAL ERRORS
C-----
C
  TYPO = .FALSE.
  DO 4010 L=3,1,-1
    IF (ANS(L:L).NE.' ') GO TO 4020
  4010 CONTINUE
  TYPO = .TRUE.
  GO TO 4040
  4020 DO 4030 K=1,L
    B = ANS(K:K)
    IF (ICHAR(B).LT.48.OR.ICHAR(B).GT.57) TYPO = .TRUE.
  4030 CONTINUE
  IF (TYPO) GO TO 4040
  DECODE(L,401,ANS) N
  IF (N.GT.NMAX) TYPO = .TRUE.
  4040 IF (TYPO) WRITE(22,410)
  RETURN
END

```

```

FUNCTION ENDATE(WHEN)
C
C   IMPLICIT INTEGER*4 (A-Z)
C
C   INTEGER*2 MCDF(12)
C   CHARACTER MM*2,DD*2,YYYY*4,WHEN*10
C
C   DATA MCDF/0,31,59,90,120,151,181,212,243,273,304,334/
C   701 FORMAT(I<L>)
C
C-----
C           CONVERT THE DATE GIVEN BY 'WHEN' INTO THE
C           NUMBER OF DAYS SINCE DECEMBER 31,1899
C-----
C
C   I = INDEX(WHEN,'/')
C   L = 2
C   IF (I.EQ.2) L = 1
C   DECODE(L,701,WHEN(1:I-1)) MONTH
C   J = INDEX(WHEN(I+1:10),'/') + I
C   K = J - I
C   L = 2
C   IF (K.EQ.2) L = 1
C   DECODE(L,701,WHEN(I+1:J-1)) DOM
C   L = 4
C   DECODE(L,701,WHEN(J+1:J+4)) YEAR
C
C   DIFF = YEAR - 1900
C   MANY = DIFF*365
C   MANY = MANY + DIFF/4 - DIFF/100 + (DIFF+300)/400
C   IF (MONTH.GT.2) THEN
C       UNLEAP = 0
C   ELSE IF (MOD(YEAR,400).EQ.0) THEN
C       UNLEAP = 1
C   ELSE IF (MOD(YEAR,100).EQ.0) THEN
C       UNLEAP = 0
C   ELSE IF (MOD(YEAR,4).EQ.0) THEN
C       UNLEAP = 1
C   ELSE
C       UNLEAP = 0
C   END IF
C   ENDATE = MANY + MCDF(MONTH) + DOM - UNLEAP
C   RETURN
C   END

```

```

FUNCTION DEDATE(MANY)
C
C   IMPLICIT INTEGER*4 (A-Z)
C
C   INTEGER*2 MCDF(12)
C   CHARACTER MM*2,DD*2,YYYY*4,DEDATE*10
C
C   DATA MCDF/0,31,59,90,120,151,181,212,243,273,304,334/
C
C   801 FORMAT(I<L>)
C
C-----
C           CONVERT NUMBER OF DAYS SINCE DECEMBER 31,1899
C                   INTO MONTH/DAY/YEAR
C-----
C
      YEAR = 1900
8010 IF (MOD(YEAR,400).EQ.0) THEN
      LEAP = 1
    ELSE IF (MOD(YEAR,100).EQ.0) THEN
      LEAP = 0
    ELSE IF (MOD(YEAR,4).EQ.0) THEN
      LEAP = 1
    ELSE
      LEAP = 0
    END IF
    DO WHILE (MANY.GT.365+LEAP)
      MANY = MANY - (365 + LEAP)
      YEAR = YEAR + 1
      GO TO 8010
    END DO
    J = 12
    DO WHILE (MANY.LE.MCDF(J))
      J = J - 1
    END DO
    MONTH = J
    DOM = MANY - MCDF(J)
    IF (DOM.EQ.0) THEN
      J = J - 1
      MONTH = J
      DOM = MCDF(J+1) - MCDF(J)
    END IF
    IF (MONTH.GT.2.AND.LEAP.EQ.1) THEN
      DOM = DOM - 1
      IF (DOM.EQ.0) THEN
        J = J - 1
        MONTH = J
        IF (MONTH.EQ.2) THEN
          DOM = 29
        ELSE
          DOM = MCDF(J+1) - MCDF(J)
        END IF
      END IF
    END IF
  
```

```

        END IF
      END IF
    END IF

```

C

```

    L = 2
    IF (MONTH/10.EQ.0) L = 1
    ENCODE(L,801,MM) MONTH
    MML = L
    L = 2
    IF (DOM/10.EQ.0) L = 1
    ENCODE(L,801,DD) DOM
    DDL = L
    L = 4
    ENCODE(L,801,YYYY) YEAR
    DEDATE = MM(1:MML)//'/'//DD(1:DDL)//'/'//YYYY
    RETURN
  END

```

FUNCTION CONVERT(A)

CHARACTER A*15

501 FORMAT(I<L>)

 CONVERT A NUMBER IN STRING FORMAT INTO ITS REAL VALUE

FRAC = 0.
 N = INDEX(A, ' ')
 IF (N.EQ.0) THEN
 N = LEN(A)
 ELSE
 N = N - 1
 END IF

INTEGER PORTION

K = INDEX(A(1:N), '.')
 IF (K.EQ.0) THEN
 L = N
 DECODE(L, 501, A) M
 X = M
 ELSE IF (K.EQ.1) THEN
 X = 0.
 ELSE
 L = K - 1
 DECODE(L, 501, A(1:L)) M
 X = M
 END IF
 IF (K.EQ.0.OR.K.EQ.N) GO TO 5020

FRACTIONAL PORTION

KP1 = K + 1
 DO 5010 J=KP1, N
 Y = ICHAR(A(J:J)) - 48
 FRAC = FRAC + Y/10**(J-K)
 010 CONTINUE

020 IF (A(1:1).EQ.'-') THEN
 X = M - FRAC
 ELSE
 X = M + FRAC
 END IF
 CONVERT = X
 RETURN
 END

DISTRIBUTION

Copies

Defense Technical Information
Center
Cameron Station
Alexandria, VA 22314

12

Commander
Naval Sea Systems Command
Attn: PMS-407E
Washington, D. C. 20362

1

Library of Congress
Attn: Gift and Exchange Division
Washington, D. C. 20540

4

Internal Distrubution

E231

9

E232

3

R44 (E. Winston)

25

U31

1

END

FILMED

9-85

DTIC